

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

J1050 U.S. PTO  
10/057654  
01/23/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 1月23日

出 願 番 号

Application Number:

特願2001-014582

出 願 人

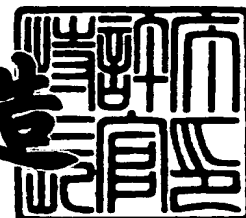
Applicant(s):

株式会社デンソー

2001年12月14日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3108451

【書類名】 特許願

【整理番号】 IP5199

【提出日】 平成13年 1月23日

【あて先】 特許庁長官殿

【国際特許分類】 H04B 7/26  
H04L 1/18

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

    【氏名】 加藤 耕治

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

    【氏名】 青野 孝之

【特許出願人】

    【識別番号】 000004260

    【氏名又は名称】 株式会社デンソー

【代理人】

    【識別番号】 100100022

    【弁理士】

    【氏名又は名称】 伊藤 洋二

    【電話番号】 052-565-9911

【選任した代理人】

    【識別番号】 100108198

    【弁理士】

    【氏名又は名称】 三浦 高広

    【電話番号】 052-565-9911

【選任した代理人】

    【識別番号】 100111578

    【弁理士】

    【氏名又は名称】 水野 史博

【電話番号】 052-565-9911

【手数料の表示】

【予納台帳番号】 038287

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 近似演算装置およびMAP復号装置

【特許請求の範囲】

【請求項1】 入力データを変数として非線形関数を近似演算する近似演算装置において、

前記入力データから、前記非線形関数を折れ線近似する傾きが2の $n$ 乗（ $n$ は整数）の直線の傾きに対応した $m$ ビット（ $m$ は自然数）のデータとその直線の切片を示すデータとを出力するデコード手段と、

前記デコード手段からの $m$ ビットのデータで前記入力データを $n$ ビットシフトするシフト手段と、

このシフト手段によってシフトされたデータと前記切片を示すデータとから前記非線形関数を示すデータを出力する手段と

を備えたことを特徴とする近似演算装置。

【請求項2】 入力データ $x$ を変数として関数 $\log(1 + e^{-x})$ を近似演算する近似演算装置において、

入力データ $x$ から、関数 $\log(1 + e^{-x})$ を折れ線近似する傾きが2の $n$ 乗（ $n$ は整数）の直線の傾きに対応した $m$ ビット（ $m$ は自然数）のデータとその直線の切片を示すデータとを出力するデコード手段と、

前記デコード手段からの $m$ ビットのデータで前記入力データを $n$ ビットシフトするシフト手段と、

このシフト手段によってシフトされたデータと前記切片を示すデータとから前記関数 $\log(1 + e^{-x})$ を示すデータを出力する手段と

を備えたことを特徴とする近似演算装置。

【請求項3】 入力データ $a$ と入力データ $b$ を変数として関数 $\log(e^a + e^b)$ を近似演算する近似演算装置において、

前記入力データ $a$ と前記入力データ $b$ のうち値が大きい方を選択して $\max(a, b)$ のデータを出力する選択手段と、

前記入力データ $a$ と前記入力データ $b$ の差の絶対値のデータから、関数 $\log(1 + e^{-|a-b|})$ を折れ線近似する傾きが2の $n$ 乗（ $n$ は整数）の直線の傾きに

対応した $m$ ビット ( $m$ は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、

前記入力データ  $a$  と前記入力データ  $b$  の差のデータを、前記デコード手段からの $m$ ビットのデータで $n$ ビットシフトするシフト手段と、

このシフト手段によってシフトされたデータと前記切片を示すデータとから関数  $\log(1 + e^{-|a-b|})$  を示すデータを出力する手段と、

前記選択手段から出力される  $\max(a, b)$  のデータと前記関数  $\log(1 + e^{-|a-b|})$  を示すデータとを加算する加算手段と、

を備えたことを特徴とする近似演算装置。

【請求項4】 前記入力データ  $a$  と前記入力データ  $b$  の差を演算して差のデータを出力する減算手段を備え、前記選択手段は、前記差のデータに基づいて前記選択を行い、前記デコード手段および前記シフト手段は、前記差のデータの絶対値に基づいてそれぞれの処理を行うことを特徴とする請求項3に記載の近似演算装置。

【請求項5】 MAP復号における各演算を指数部のみで行うlog#BCJRアルゴリズムを用いてMAP復号を行うMAP復号装置において、

前方繰り返し演算による状態確率の演算および後方繰り返し演算による状態確率の演算を、入力データ  $a$  と入力データ  $b$  を変数として関数  $\log(e^a + e^b)$  の近似演算にて行うものであって、前記関数  $\log(e^a + e^b)$  の近似演算のために、前記入力データ  $a$  と前記入力データ  $b$  のうち値が大きい方を選択して  $\max(a, b)$  のデータを出力する選択手段と、

前記入力データ  $a$  と前記入力データ  $b$  の差の絶対値のデータから、関数  $\log(1 + e^{-|a-b|})$  を折れ線近似する傾きが2の $n$ 乗 ( $n$ は整数) の直線の傾きに対応した $m$ ビット ( $m$ は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、

前記入力データ  $a$  と前記入力データ  $b$  の差のデータを、前記デコード手段からの $m$ ビットのデータで $n$ ビットシフトするシフト手段と、

このシフト手段によってシフトされたデータと前記切片を示すデータとから関数  $\log(1 + e^{-|a-b|})$  を示すデータを出力する手段と、

前記選択手段から出力される  $\max(a, b)$  のデータと前記関数  $\log(1 + e^{-|a-b|})$  を示すデータとを加算する加算手段と、

を備えたことを特徴とするMAP復号装置。

【請求項6】 前記入力データaと前記入力データbの差を演算して差のデータを出力する減算手段を備え、前記選択手段は、前記差のデータに基づいて前記選択を行い、前記デコード手段および前記シフト手段は、前記差のデータの絶対値に基づいてそれぞれの処理を行うことを特徴とする請求項5に記載のMAP復号装置。

# 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明は、非線形関数の近似演算装置、関数  $\log(1 + e^{-x})$  の近似演算装置、関数  $\log(e^a + e^b)$  の近似演算装置、およびMAP復号装置に関する。

## 【0002】

## 【従来の技術および発明が解決しようとする課題】

従来、畳込み符号とインタリーブとを組合せたターボ符号に変換して送信された符号化データを受信し、その誤りを訂正して復号を行うターボデコーダが提案されている。

## 【0003】

このターボデコーダの構成を図8に示す。このターボデコーダは、2つの軟出力復号器101、102と、2つのインターリーバ103、104と、デインターリーバ105と、硬判定器106から構成された公知のものである。ターボデコーダの構成要素である軟出力復号器101、102の復号としては、MAP（最大事後確率：Maximum A Posteriori Probability）復号が用いられる。

## 【0004】

図9に、MAP復号の演算処理を示す。入力データが1、0のいずれであるかの確率PDF（受信信号の確率密度関数：probability density function）を演算するステップ201、トレリスの各枝（ブランチ）の確率（状態遷移確率） $\gamma$  を演算するステップ202、トレリスをデータの先頭から演算して個々のビット

の各状態の確率（前方繰り返し演算による状態確率） $\alpha$ を演算するステップ203、トレリスをデータの最後尾から演算して $\alpha$ と同様に個々のビットの各状態の確率（後方繰り返し演算による状態確率） $\beta$ を演算するステップ204、先に求めた $\gamma$ 、 $\alpha$ 、 $\beta$ からトレリスの各ビットが1、0のいずれであるかの確率（情報シンボルの尤度） $\lambda$ を演算するステップ205、 $\lambda$ で得られた確率を尤度としてフィードバックするために事前尤度（LDK）を演算するステップ206、および $\lambda$ で得られた確率を硬判定して入力データを復号するステップ207からなる。

#### 【0005】

図10に、このMAP復号の各演算処理を演算器で行う場合のブロック構成を示す。図9に示したPDF、 $\gamma$ 、 $\alpha$ 、 $\beta$ 、 $\lambda$ 、LDKは、式を簡略化すれば、図10に示す演算式を各々の演算器で演算していることになる。このようなMAP復号の演算式をそのままハードウェアあるいはソフトウェアで実現すると、各中間結果（データ）のビット数が大きくなり、非実現的なものになってしまう。

#### 【0006】

そこで、MAP復号を、より小さな演算及び小型な回路で実現するために、様々なアルゴリズムが提案されている。その一例として、log#BCJRアルゴリズム（S. Bebedetto et. al "Soft-output decoding of parallel concatenated convolutional codes", ICC 96）は、演算量を減らしながら復号能力を維持できるものとして広く用いられている。

#### 【0007】

log#BCJRアルゴリズムでは、図10に示した各演算を指数部のみで演算する。具体的には、対数をとることにより、PDFは、 $-(x-1)^2/2\sigma$ で求められる（ $x$ は入力データ、 $\sigma$ は分散値を示す）。また、 $p d f \cdot p d f \cdot p r b$ は指数部を示しているので、 $\gamma$ は $p d f + p d f + p r b$ のように加算で求められる、 $\lambda$ は（分子-分母）の減算で求められる。

#### 【0008】

しかし、 $\alpha$ 、 $\beta$ に関しては、新たに $\log(e^a + e^b)$ の演算が生じてしまう。すなわち、 $\alpha$ は、 $\alpha \cdot \gamma + \alpha \cdot \gamma$ の演算で求められるが、具体的には、1つ前

の2つのトレリスの状態 $\alpha_1$ 、 $\alpha_2$ と、 $\alpha_1$ 、 $\alpha_2$ からの枝 $\gamma_1$ 、 $\gamma_2$ によって、 $\alpha_1 \cdot \gamma_1 + \alpha_2 \cdot \gamma_2$ の演算式で求められる。ここで、 $\alpha \cdot \gamma + \alpha \cdot \gamma$ の各項は、指数関数で表されているため、 $\alpha \cdot \gamma + \alpha \cdot \gamma$ の指数部は、 $\log(e^a + e^b)$ の演算で求められる。同様に、 $\beta$ は、 $\beta \cdot \gamma + \beta \cdot \gamma$ の演算で求められるが、具体的には、1つ前の2つのトレリスの状態 $\beta_1$ 、 $\beta_2$ と、 $\beta_1$ 、 $\beta_2$ からの枝 $\gamma_3$ 、 $\gamma_4$ によって、 $\beta_1 \cdot \gamma_3 + \beta_2 \cdot \gamma_4$ の演算式で求められる。ここで、 $\beta \cdot \gamma + \beta \cdot \gamma$ の各項は、指数関数で表されているため、 $\beta \cdot \gamma + \beta \cdot \gamma$ の指数部は、 $\log(e^a + e^b)$ の演算で求められる。

【0009】

そこで、log#BCJRアルゴリズムでは、 $\log(e^a + e^b)$ を数式1に変形して演算することで、演算を簡略化している。

【0010】

【数1】

$$\max(a, b) + \log(1 + e^{-|a-b|})$$

すなわち、 $a > b$ の場合、 $\log(e^a + e^b)$ は、 $\log e^a (1 + e^{b-a}) = a + \log(1 + e^{b-a})$ と変形され、 $b > a$ の場合、 $\log(e^a + e^b)$ は、 $\log e^b (1 + e^{a-b}) = b + \log(1 + e^{a-b})$ と変形される。このことから、 $\log(e^a + e^b)$ は、数式1のように変形することができる。

【0011】

ここで、数式1の第2項 $\log(1 + e^{-|a-b|})$ をどのようにして求めるかが問題となる。2000年電子情報通信学会総合大会の「W-CDMAターボ復号器のVLSI化と評価」(B-5-26)では、テーブルルックアップにより $\log(1 + e^{-|a-b|})$ を求めることが提案されている。

【0012】

図11に、テーブルルックアップを用いて、数式1を演算する場合の構成を示す。2つの入力データ $a$ 、 $b$ が減算器301に入力され、この減算器301で $a$ 、 $b$ の差( $a - b$ )が求められる。また、入力データ $a$ 、 $b$ は選択器302に入力され、この選択器302で、減算器301から出力される差( $a - b$ )の符号に応じて $\max(a, b)$ が求められる。また、減算器301から出力される差



( $a - b$ ) は、テーブルルックアップ 3 0 3 に入力され、このテーブルルックアップ 3 0 3 で、差 ( $a - b$ ) に応じた  $\log(1 + e^{-|a-b|})$  の値がテーブルから求められる。そして、加算器 3 0 4 で、 $\max(a, b)$  と  $\log(1 + e^{-|a-b|})$  の値が加算され、数式 1 の出力  $y$  が得られる。

## 【 0 0 1 3 】

しかし、このように数式 1 の第 2 項  $\log(1 + e^{-|a-b|})$  をテーブルルックアップ 3 0 3 で実現した場合、復号能力を維持するためには、テーブルを詳細に用意する必要があり、回路規模あるいは演算量の削減を妨げる要因となる。

## 【 0 0 1 4 】

また、一般に、 $\log(1 + e^{-x})$  とか、その他の非線形関数をデジタル処理で求める場合にも、テーブルルックアップを用いることが考えられるが、テーブルルックアップを用いたのでは、テーブルを詳細に用意する必要があり、回路規模ないし演算量の削減を妨げる要因となる。

## 【 0 0 1 5 】

本発明は上記問題に鑑みたもので、log#BCJR アルゴリズムにおける数式 1 の第 2 項を、より小型な回路あるいは少ない演算量で演算できるようにすることを第 1 の目的とする。

## 【 0 0 1 6 】

また、 $\log(1 + e^{-x})$ 、およびその他の非線形関数において、より小型な回路あるいは少ない演算量で演算できるようにすることを第 2 の目的とする。

## 【 0 0 1 7 】

## 【課題を解決するための手段】

上記目的を達成するため、請求項 1 に記載の発明では、入力データを変数として非線形関数を近似演算する近似演算装置において、前記入力データから、前記非線形関数を折れ線近似する傾きが 2 の  $n$  乗 ( $n$  は整数) の直線の傾きに対応した  $m$  ビット ( $m$  は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、前記デコード手段からの  $m$  ビットのデータで前記入力データを  $n$  ビットシフトするシフト手段と、このシフト手段によってシフトされたデータと前記切片を示すデータとから前記非線形関数を示すデータを出力する手段とを

備えたことを特徴としている。

【0018】

このことにより、非線形関数を、より小型な回路あるいは少ない演算量で演算することができる。

【0019】

請求項2に記載の発明では、入力データ  $x$  を変数として関数  $\log(1 + e^{-x})$  を近似演算する近似演算装置において、入力データ  $x$  から、関数  $\log(1 + e^{-x})$  を折れ線近似する傾きが2の  $n$  乗 ( $n$  は整数) の直線の傾きに対応した  $m$  ビット ( $m$  は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、前記デコード手段からの  $m$  ビットのデータで前記入力データを  $n$  ビットシフトするシフト手段と、このシフト手段によってシフトされたデータと前記切片を示すデータとから前記関数  $\log(1 + e^{-x})$  を示すデータを出力する手段とを備えたことを特徴としている。

【0020】

このことにより、関数  $\log(1 + e^{-x})$  を、より小型な回路あるいは少ない演算量で演算することができる。

【0021】

請求項3に記載の発明では、入力データ  $a$  と入力データ  $b$  を変数として関数  $\log(e^a + e^b)$  を近似演算する近似演算装置において、前記入力データ  $a$  と前記入力データ  $b$  のうち値が大きい方を選択して  $\max(a, b)$  のデータを出力する選択手段と、前記入力データ  $a$  と前記入力データ  $b$  の差の絶対値のデータから、関数  $\log(1 + e^{-|a-b|})$  を折れ線近似する傾きが2の  $n$  乗 ( $n$  は整数) の直線の傾きに対応した  $m$  ビット ( $m$  は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、前記入力データ  $a$  と前記入力データ  $b$  の差のデータを、前記デコード手段からの  $m$  ビットのデータで  $n$  ビットシフトするシフト手段と、このシフト手段によってシフトされたデータと前記切片を示すデータとから関数  $\log(1 + e^{-|a-b|})$  を示すデータを出力する手段と、前記選択手段から出力される  $\max(a, b)$  のデータと前記関数  $\log(1 + e^{-|a-b|})$  を示すデータとを加算する加算手段とを備えたことを特徴としている。

## 【0022】

このことにより、関数  $\log(e^a + e^b)$  を、より小型な回路あるいは少ない演算量で演算することができる。

## 【0023】

この場合、請求項4に記載の発明のように、入力データ  $a$  と前記入力データ  $b$  の差を演算して差のデータを出力する減算手段を備え、前記選択手段が、前記差のデータに基づいて前記選択を行い、前記デコード手段および前記シフト手段が、前記差のデータの絶対値に基づいてそれぞれの処理を行うようにすることができる。

## 【0024】

請求項5に記載の発明では、MAP復号における各演算を指数部のみで行う  $\log$ #BCJRアルゴリズムを用いてMAP復号を行うMAP復号装置において、前方繰返し演算による状態確率の演算および後方繰返し演算による状態確率の演算を、入力データ  $a$  と入力データ  $b$  を変数として関数  $\log(e^a + e^b)$  の近似演算にて行うものであって、前記関数  $\log(e^a + e^b)$  の近似演算のために、前記入力データ  $a$  と前記入力データ  $b$  のうち値が大きい方を選択して  $\max(a, b)$  のデータを出力する選択手段と、前記入力データ  $a$  と前記入力データ  $b$  の差の絶対値のデータから、関数  $\log(1 + e^{-|a-b|})$  を折れ線近似する傾きが2の  $n$  乗 ( $n$  は整数) の直線の傾きに対応した  $m$  ビット ( $m$  は自然数) のデータとその直線の切片を示すデータとを出力するデコード手段と、前記入力データ  $a$  と前記入力データ  $b$  の差のデータを、前記デコード手段からの  $m$  ビットのデータで  $n$  ビットシフトするシフト手段と、このシフト手段によってシフトされたデータと前記切片を示すデータとから関数  $\log(1 + e^{-|a-b|})$  を示すデータを出力する手段と、前記選択手段から出力される  $\max(a, b)$  のデータと前記関数  $\log(1 + e^{-|a-b|})$  を示すデータとを加算する加算手段とを備えたことを特徴としている。

## 【0025】

このことにより、 $\log$ #BCJRアルゴリズムを用いたMAP復号処理において、 $\max(a, b) + \log(1 + e^{-|a-b|})$  の演算を、より小型な回路あるいは少

ない演算量で演算することができる。

【0026】

この場合、請求項6に記載の発明のように、入力データaと前記入力データbの差を演算して差のデータを出力する減算手段を備え、前記選択手段が、前記差のデータに基づいて前記選択を行い、前記デコード手段および前記シフト手段が、前記差のデータの絶対値に基づいてそれぞれの処理を行うようにすることができる。

【0027】

なお、上記した非線形関数、関数 $\log(1+e^{-x})$ 、関数 $\log(e^a+e^b)$ の近似は、傾きが2のn乗（nは整数）の直線を隣接する直線との交点を区切りとしてつなぎ合わせ、関数との誤差を最小とするように折れ線グラフで近似することの他、傾きが2のn乗（nは整数）の接線を隣接する接線との交点を区切りとしてつなぎ合わせ、折れ線グラフで近似するようにして求めることができる。

【0028】

【発明の実施の形態】

（第1実施形態）

図1に、非線形関数を近似演算する装置の構成を示す。この装置は、図2に示すような非線形関数を直線近似して、入力データxに対する出力データyを求めるものである。

【0029】

ここで、直線近似するための各直線は、図2に示す非線形関数において、傾きが4、2、1、0、0.5、0.25のような2のn乗（nは整数）となる各直線を図のように沿わせることにより求められる。隣り合った直線の交点を区切りとして、各直線を図のように折れ線グラフとしてつなぐことで非線形関数を近似する。この場合、非線形関数のグラフ上の任意のサンプル点における非線形関数のy座標と、切片を任意に変更できる当該直線上のy座標との差分を誤差として、この誤差を最小とするように切片が決定される。このサンプル点の数を増やせば、より精度の高い近似が可能となる。

## 【0030】

このようにして求められた直線の傾きAと切片Bから、出力yは、 $y = A \cdot x + B$ で求められる。

## 【0031】

デジタルの入力データxは、デコーダ11に入力される。デコーダ11は、入力データxから、上記のようにして求めておいた直線の傾きAを示す2のn乗（nは整数）の直線の傾きに対応したmビット（mは自然数）のデータと切片Bのデータをそれぞれ出力する。シフター12は、入力データxをmビットのデータでnビットだけシフトして、 $A \cdot x$ のデータを出力する。そして、加算器13にて、 $A \cdot x$ のデータと切片Bのデータが加算され、出力データyが得られる。

## 【0032】

なお、上記した例では、非線形関数のグラフ上の任意のサンプル点における非線形関数のy座標と、切片を任意に変更できる当該直線上のy座標との差分を誤差として、この誤差を最小とするように切片を決定するものを示したが、図3に示すように、非線形関数に対し傾きが4、2、1、0、0.5、0.25のような2のn乗（nは整数）となる各接線を、図のように隣り合った直線の交点を区切りとして、各直線を折れ線グラフとしてつなぐことで非線形関数を近似するようにしてもよい。具体的には、各接線は、非線形関数を微分し、この微分式が各傾き（4、2、1、0、0.5、0.25）になるようにして求められる。

## 【0033】

従来のテーブルルックアップによる近似では、図4に示すように、非線形関数を図2、図3と同程度に近似するためには、非常に沢山の参照点を必要とする。このような参照点の増加は、回路規模の増加につながるという問題がある。また、参照点を減らし、参照点間を直線で近似する方式を用いた場合には、精度的には図1と同等に成り得るが、各参照点間を結ぶ直線の傾きは任意であるため、乗算及び除算を必要とし、回路規模及び演算量が増大するという問題がある。図2、図3に示す近似によれば、各直線の区切りを少なくでき、また各直線の傾きが2のn乗（nは整数）で表されるため、ビットシフトで演算でき、回路規模及び演算量を削減することができる。

## 【0034】

また、非線形関数として、 $\log(1 + e^{-x})$  とした場合にも、上記と同様の直線近似にて、入力データ  $x$  に対する出力データ  $y$  を得ることができる。この場合、図5に、図2と同様、傾きが2の  $n$  乗（但し、 $n$  は負の整数で、 $-0.5$ 、 $-0.25$ 、 $-0.125$ 、 $-0.0625$ 、 $-0.03125 \dots$ ）となる直線と非線形関数との誤差を最小とするように切片を決定するものを示し、図6に、図3と同様、非線形関数に対し傾きが2の  $n$  乗（但し、 $n$  は負の整数で、 $-0.5$ 、 $-0.25$ 、 $-0.125$ 、 $-0.0625$ 、 $-0.03125 \dots$ ）となる各接線を用い、各直線を折れ線グラフとしてつなぐことで非線形関数を近似するものを示す。

## 【0035】

このように非線形関数として  $\log(1 + e^{-x})$  とした場合も、図1に示す構成によって、入力データ  $x$  に対する出力データ  $y$  を得ることができる。但し、この例では、傾きが2の  $n$  乗で指数部  $n$  が必ず負の整数になる。このため、デコーダ11が、傾きが2の  $n$  乗の指数部  $n$  を正の整数とした  $m$  ビットのデータを入力するようにすれば、シフター12は入力データ  $x$  を  $n$  ビットだけシフトし、それを加算器13の代わりに用いた減算器にて切片Bのデータから減算するようにすれば、 $\log(1 + e^{-x})$  の値を示すデータを得ることができる。

## (第2実施形態)

図7に、MAP復号における  $\alpha$  演算を行う演算器、 $\beta$  演算を行う演算器のそれぞれにおいて、log#BCJRアルゴリズムに従い、数式1に示す  $\max(a, b) + \log(1 + e^{-|a-b|})$  の演算を行う構成を示す。

## 【0036】

図7に示す構成において、2つのデジタルの入力データ  $a$ 、 $b$  が減算器21に入力され、この減算器21で  $a$ 、 $b$  の差  $(a - b)$  が求められる。また、入力データ  $a$ 、 $b$  は選択器22に入力され、この選択器22で、減算器21から出力される差  $(a - b)$  の符号に応じて  $\max(a, b)$  が求められる。また、減算器21から出力される差  $(a - b)$  は、デコーダ23に入力される。

## 【0037】

ここで、 $\log(1 + e^{-|a-b|})$  は、 $|a-b|$  を  $x$  とすれば、図5、図6に示す実施形態と同じ形になるため、まず、絶対値回路（絶対値検出手段）27で、減算器21から出力される差  $(a-b)$  の絶対値を求める。そして、 $(a-b)$  の絶対値を入力データ  $x$  として、デコーダ23およびシフター24のそれぞれの処理が行われる。すなわち、デコーダ23は、入力データ  $x$  を用いて、直線の傾き  $A$  を示す2の  $n$  乗（ $n$  は負の整数）の指数部  $n$  を正の整数とした  $m$  ビットのデータと切片  $B$  のデータをそれぞれ出力する。シフター24は、入力データ  $x$  を  $n$  ビットだけシフトして、 $A \cdot x$  のデータを出力する。

【0038】

そして、減算器25にて、切片  $B$  のデータから  $A \cdot x$  のデータが減算される。この減算結果は、 $\log(1 + e^{-|a-b|})$  の値を示す。

【0039】

最後に、加算器26で、 $\log(1 + e^{-|a-b|})$  の値と選択器22から出力される  $\max(a, b)$  が加算され、 $y = \max(a, b) + \log(1 + e^{-|a-b|})$  が出力される。

【0040】

上記した構成により、 $\log$ #BCJRアルゴリズムにおける  $\max(a, b) + \log(1 + e^{-|a-b|})$  の演算を行うことができる。このような演算を用いるMAP復号器は、複数の演算器により図10に示すようなハードウェアで実現される他、図9に示すようなソフトウェアで実現することができる。ソフトウェアで実現した場合、上記した処理は、図9の演算  $\alpha$ 、演算  $\beta$  の処理（ステップ203、204）で用いられる。また、このようなMAP復号器は、例えば図8に示すターボデコーダの軟出力復号器101、102として用いられる。

【0041】

また、 $\alpha$ 、 $\beta$  以外に  $\lambda$  についても  $\log(e^a + e^b)$  の演算が必要であれば、上記と同様の演算処理を行うようにしてもよい。

【図面の簡単な説明】

【図1】

非線形関数を近似演算する装置の構成を示す図である。

【図 2】

非線形関数を直線近似する方法を説明するための図である。

【図 3】

非線形関数を直線近似する他の方法を説明するための図である。

【図 4】

従来のテーブルルックアップによる近似を説明するための図である。

【図 5】

$\log(1 + e^{-x})$  を直線近似する方法を説明するための図である。

【図 6】

$\log(1 + e^{-x})$  を直線近似する他の方法を説明するための図である。

【図 7】

$\log\#BCJR$  アルゴリズムを用いた MAP 復号処理において、 $\max(a, b) + \log(1 + e^{-|a-b|})$  の演算を行う装置の構成を示す図である。

【図 8】

ターボデコーダの構成を示す図である。

【図 9】

MAP 復号の演算処理を示す図である。

【図 10】

MAP 復号の各演算処理を演算器で行う場合のブロック構成を示す図である。

【図 11】

テーブルルックアップを用いて、 $\max(a, b) + \log(1 + e^{-|a-b|})$  の演算を行う装置の構成を示す図である。

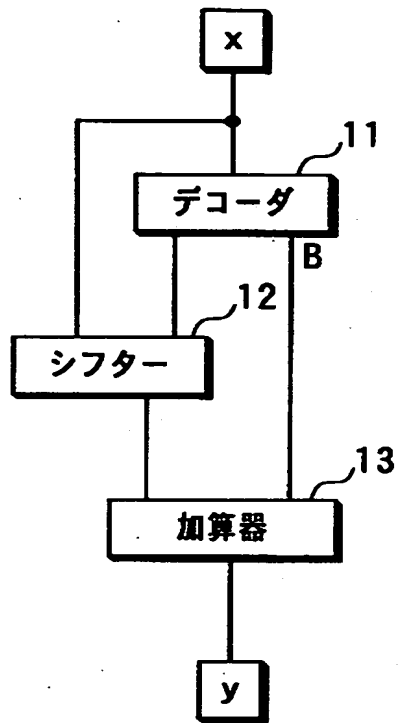
【符号の説明】

11…デコーダ、12…シフター、13…加算器、21…減算器、  
22…選択器、23…デコーダ、24…シフター、25…減算器、  
26…加算器、27…絶対値回路。

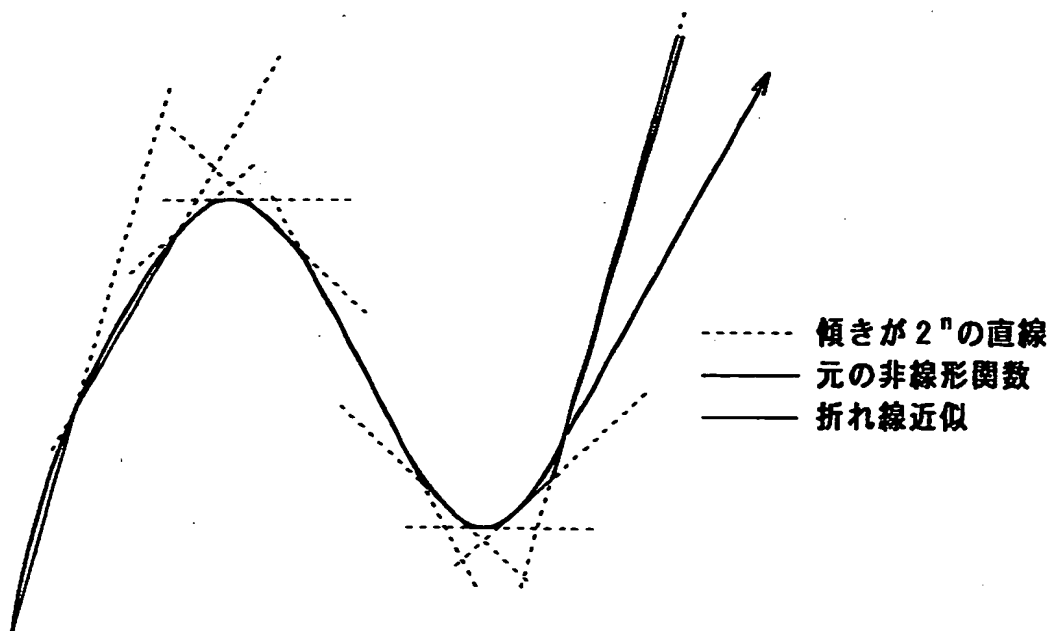


【書類名】 図面

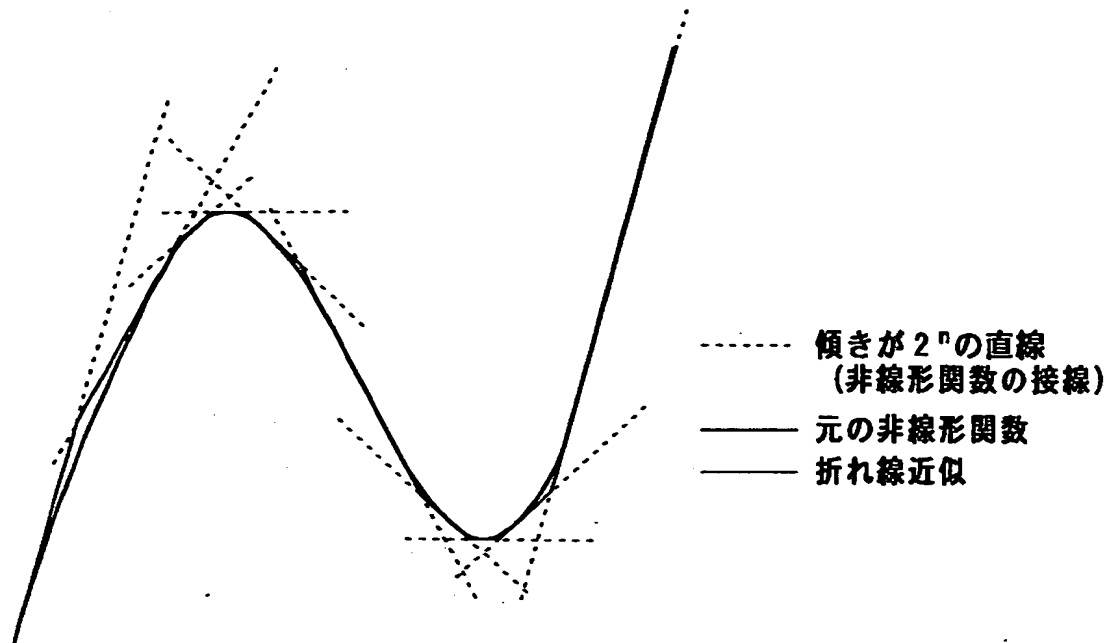
【図 1】



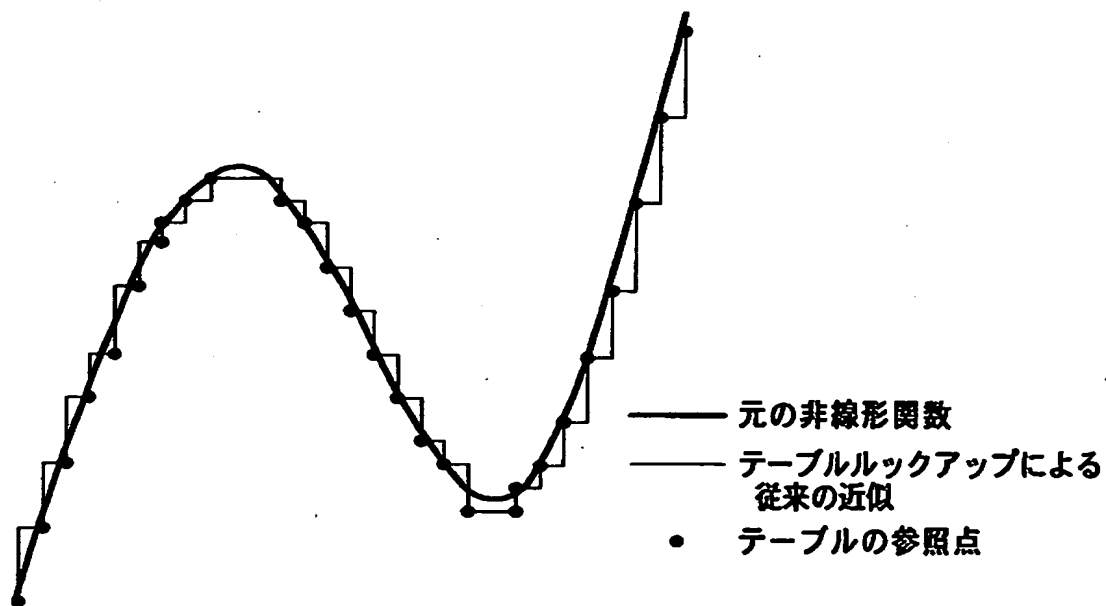
【図 2】



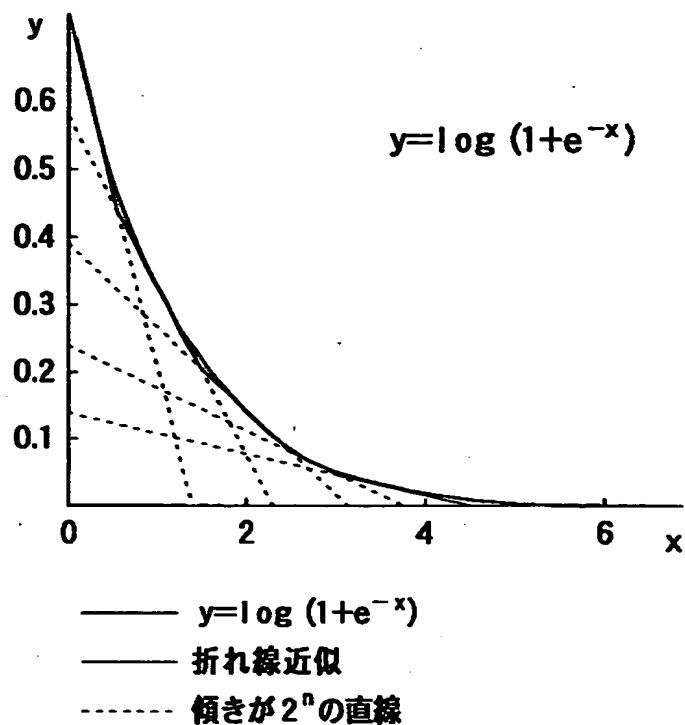
【図3】



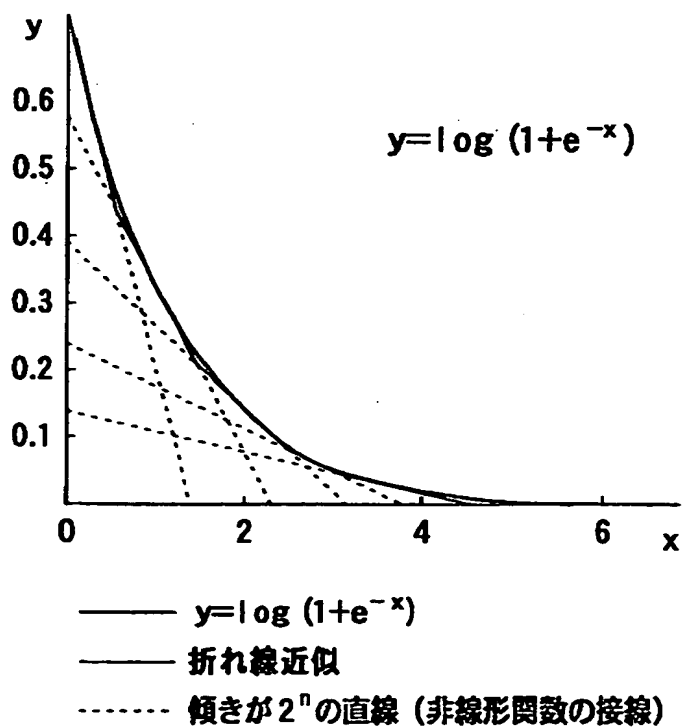
【図4】



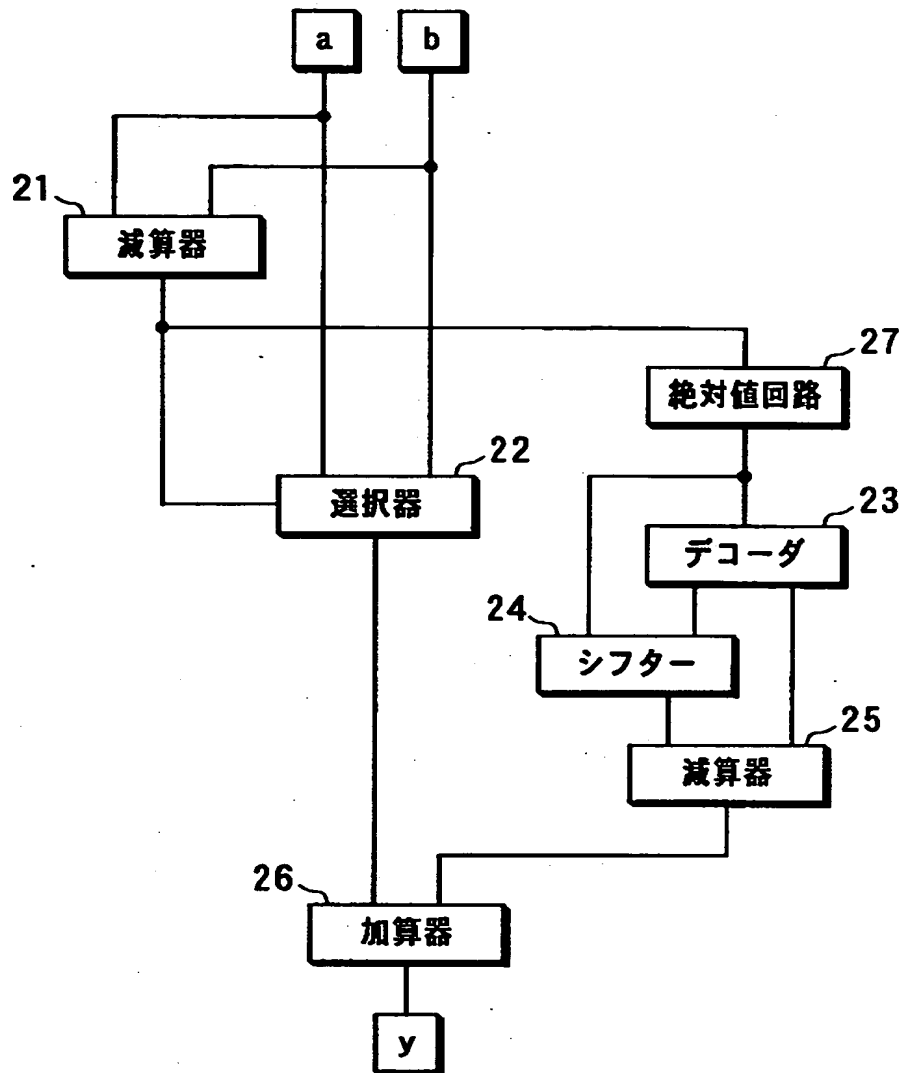
【図5】



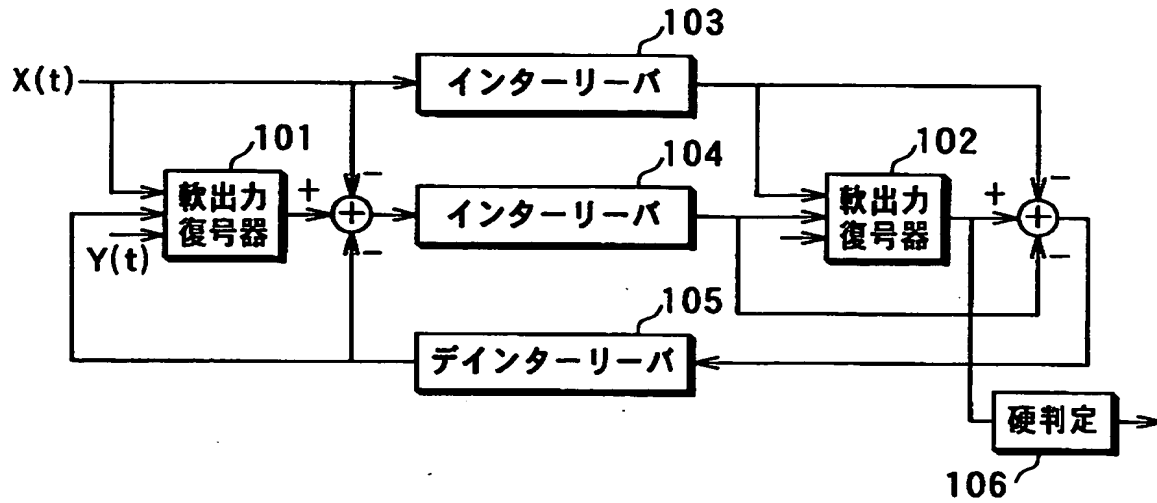
【図6】



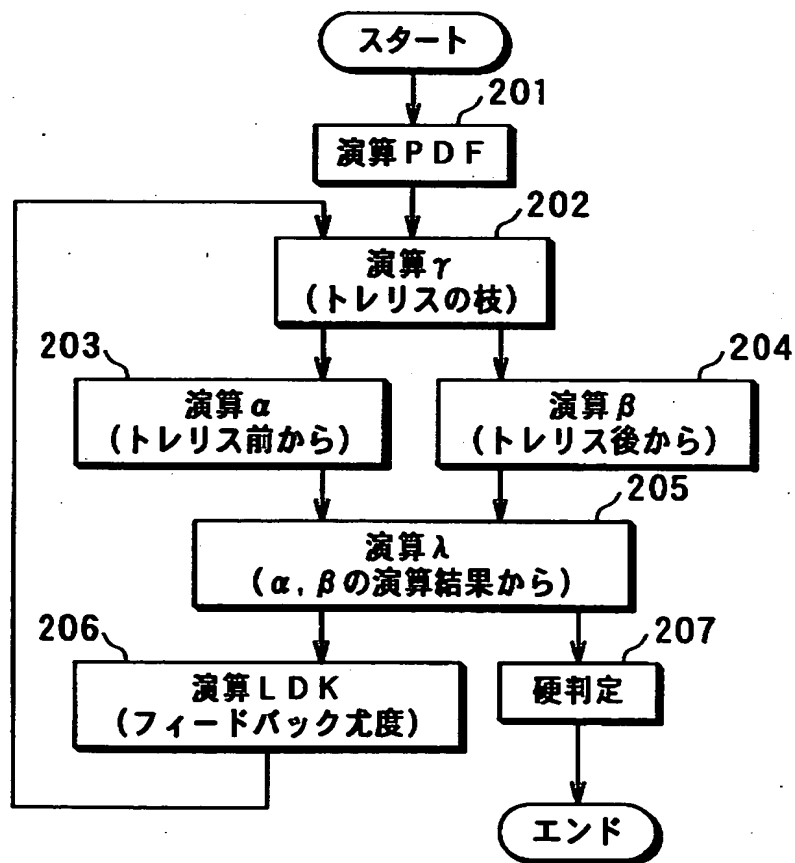
【図 7】



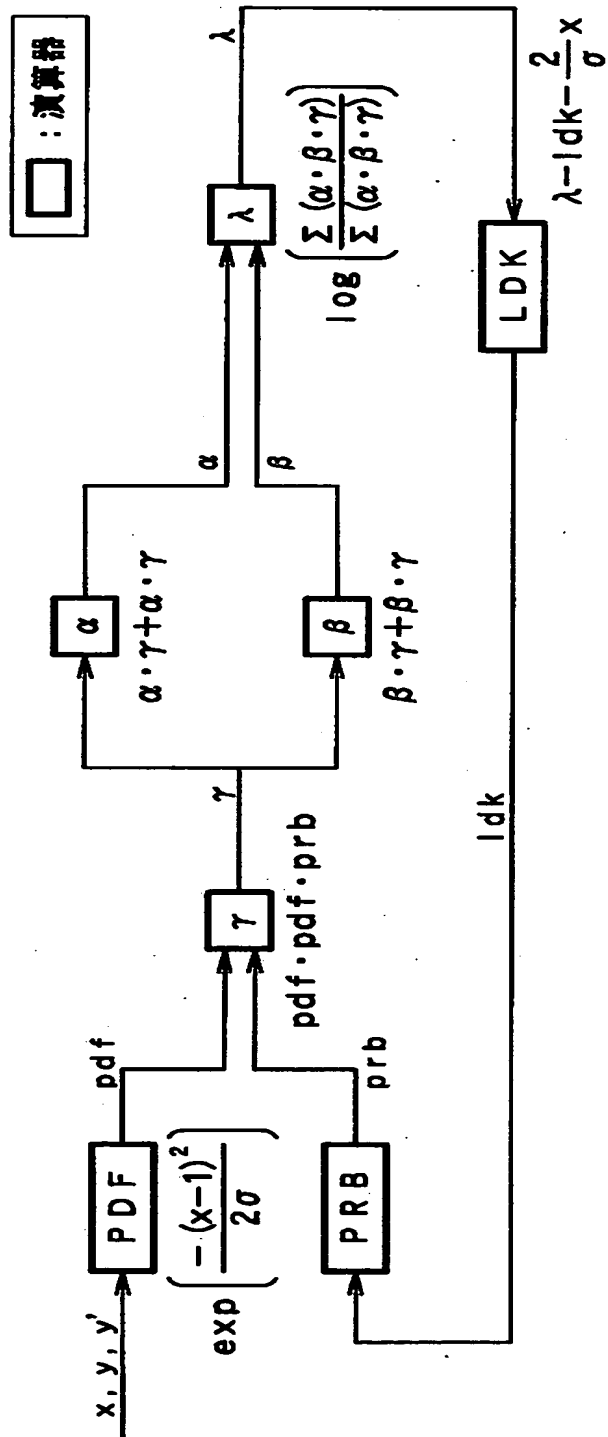
【図 8】



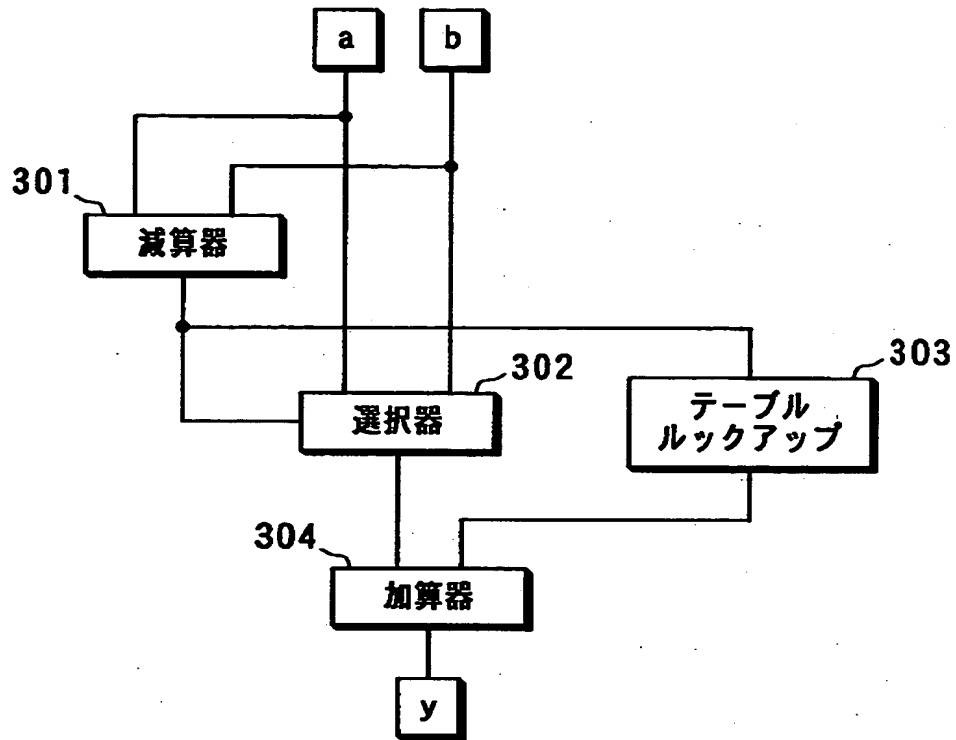
【図 9】



【図10】



【図 11】



【書類名】 要約書

【要約】

【課題】  $\log\#BCJR$  アルゴリズムを用いた MAP 復号処理において、 $\max(a, b) + \log(1 + e^{-|a-b|})$  をより少ない演算量で演算する。

【解決手段】 減算器 21 は、入力データ  $a$ 、 $b$  の差  $(a - b)$  を求める。選択器 22 は、入力データ  $a$ 、 $b$  から  $\max(a, b)$  を求める。絶対値回路 27 は、差  $(a - b)$  の絶対値を求める。これを入力データ  $x$  とし、デコーダ 23 は、入力データ  $x$  を用いて、 $\log(1 + e^{-|a-b|})$  を直線近似した直線の傾き  $A$  を示す 2 の  $n$  乗 ( $n$  は負の整数) の指数部  $n$  を正の整数とした  $m$  ビットのデータと、切片  $B$  のデータを出力する。シフター 24 は、入力データ  $x$  を  $n$  ビットだけシフトして、 $A \cdot x$  のデータを出力する。減算器 25 は、切片  $B$  のデータから  $A \cdot x$  のデータを減算し、 $\log(1 + e^{-|a-b|})$  の値を示すデータを出力する。加算器 26 は、 $\log(1 + e^{-|a-b|})$  の値と  $\max(a, b)$  を加算する。

【選択図】 図 7



出 願 人 履 歴 情 報

識別番号 [000004260]

1. 変更年月日	1996年10月 8日
[変更理由]	名称変更
住 所	愛知県刈谷市昭和町1丁目1番地
氏 名	株式会社デンソー